

# Accurate reconstruction of insertion-deletion histories by statistical phylogenetics

Oscar Westesson<sup>1</sup>, Gerton Lunter<sup>2</sup>, Benedict Paten<sup>3</sup>, Ian Holmes<sup>1,\*</sup>

**1** UC Berkeley and UCSF Graduate Program in Bioengineering, University of California, Berkeley, CA, USA;

**2** Wellcome Trust Center for Human Genetics, Oxford, Oxford, UK;

**3** Baskin School of Engineering, UC Santa Cruz, Santa Cruz, CA, USA

\* E-mail: [protpal@postbox.biowiki.org](mailto:protpal@postbox.biowiki.org)

## Abstract

The Multiple Sequence Alignment (MSA) is a computational abstraction that represents a partial summary either of indel history, or of structural similarity. Taking the former view (indel history), it is possible to use formal automata theory to generalize the phylogenetic likelihood framework for finite substitution models (Dayhoff's probability matrices and Felsenstein's pruning algorithm) to arbitrary-length sequences. In this paper, we report results of a simulation-based benchmark of several methods for reconstruction of indel history. The methods tested include a relatively new algorithm for statistical marginalization of MSAs that sums over a stochastically-sampled ensemble of the most probable evolutionary histories. For mammalian evolutionary parameters on several different trees, the single most likely history sampled by our algorithm appears less biased than histories reconstructed by other MSA methods. The algorithm can also be used for alignment-free inference, where the MSA is explicitly summed out of the analysis. As an illustration of our method, we discuss reconstruction of the evolutionary histories of human protein-coding genes.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Results</b>	<b>6</b>
2.1	Computational reconstruction of simulated histories . . . . .	6
2.1.1	Simulation model parameters . . . . .	7
2.1.2	Indel rate estimates . . . . .	8
2.2	Reconstructed indel histories of human genes . . . . .	11
<b>3</b>	<b>Discussion</b>	<b>13</b>
<b>4</b>	<b>Methods</b>	<b>16</b>
4.1	Felsenstein’s algorithm for indel models . . . . .	16
4.2	Transducer definitions and lemmas . . . . .	18
4.3	The phylogenetic likelihood . . . . .	21
4.4	Alignment envelopes . . . . .	23
4.5	OPTIC data analysis . . . . .	25
<b>5</b>	<b>Figures</b>	<b>26</b>
<b>6</b>	<b>References</b>	<b>29</b>
<b>A</b>	<b>Simulation parameters and setup</b>	<b>33</b>
<b>B</b>	<b>Supplemental Figures: simulated data analysis</b>	<b>37</b>
<b>C</b>	<b>Supplemental Figures: OPTIC analysis</b>	<b>38</b>

## 1 Introduction

The Multiple Sequence Alignment (MSA), indispensable to computational sequence analysis, represents a hypothetical claim about the homology between sequences. MSAs have many different uses, but the underlying hypothesis can often be classified as a claim either of *structural* homology (the 3D structures align in a particular way) or of *evolutionary* homology (the sequences are related by a particular history on a given phylogenetic tree). These types of hypothesis are similar, but with subtle (and important) distinctions: at the residue level, a claim of evolutionary homology (direct shared descent) is far stronger than a claim of structural homology (same approximate fold). Furthermore, both types of MSA—evolutionary and structural—typically only represent *summaries* of the respective homologies: some fine detail is often omitted. For example, an evolutionary MSA may—or may not—include the ancestral sequences at internal nodes of the underlying tree.

Structural and evolutionary MSAs are often conflated, but they have quite different applications. For example, a common use for a structural MSA is *template-based structure prediction*, where a query sequence is aligned to a target of known structure; the success of this prediction reflects the number of query-template residues correctly aligned [1]. By way of contrast, a common application for an evolutionary MSA is to identify regions or sites under selection, the success of which depends on accurate reconstruction of the evolutionary history [2, 3].

Evaluation of alignment methods is typically done with implicit regard for the structural interpretation. Many benchmarks have used metrics based on the Sum of Pairs Score (SPS) [4]. In the situation that a query-template pairwise alignment is randomly picked out of the MSA, the SPS effectively estimates the proportion of homologous residues that are correctly identified. Several

alignment methods attempt to maximize the posterior expectation of SPS or similar metrics. This appears to improve accuracy, particularly when measured with reference to structural homology. However, it does not automatically confer *evolutionary* accuracy — a correct reconstruction of the evolutionary history of the sequences.

Several studies suggest that multiple alignment for evolutionary purposes is still a highly uncertain procedure [5], and that errors therein may significantly bias analyses of evolutionary effects [6–11]. A useful component of these studies is simulation of genetic sequence evolution [6], which appears to better indicate evolutionary accuracy than benchmarks derived from protein structure alignments. Simulations can be made quite realistic given the abundance of comparative sequence data [12].

The current state-of-the-art in phylogenetic alignment software is a choice between (on the one hand) programs that lack explicit models of the underlying evolutionary process, and so are not framed as statistical inference problems [6], and (on the other hand) Bayesian Markov chain Monte Carlo (MCMC) methods, which are statistically exact but prohibitively slow [13, 14].

A telling observation is that while substitution rate is routinely measured from MSAs and used as an indicator of natural selection, there is relatively little analogous use of indel rate. As we report here, it seems highly likely that even if indel rate is a useful evolutionary signal (which is eminently plausible), the present alignment methods distort measurements of this rate so far as to make it meaningless (see especially Figure 1).

In this paper, we frame phylogenetic sequence alignment as an approximate maximum likelihood (ML) inference. Our inference algorithm assumes that the tree is known, requiring a separate tree estimation protocol. While this is a strong assumption, it is in principle shared among all progressive aligners

(e.g. PRANK [15], Muscle [16], ClustalW [17], MAFFT [18]). The alignment-marginalized likelihoods reported by our algorithm allow for statistical tests between alternative trees, and the functionality to estimate an initial alignment and guide tree from unaligned sequences exists elsewhere in the DART package. Our framing uses automata-theoretic methods from computational linguistics to unify several previously-disjoint areas of bioinformatics: Felsenstein’s pruning algorithm for the phylogenetic likelihood function [19], progressive multiple sequence alignment [20], and alignment ensemble representation using partial order graphs [21]. Our algorithm may be viewed as a stochastic generalization of pruning to infinite state spaces: it retains the linear time and memory complexity of pruning ( $\mathcal{O}(NL)$  for  $N$  sequences of length  $L$ ), while moderating the biasing effect of the MSA. The algorithmic details of our method are outlined briefly in the Methods, and in more complete, mathematically precise terms (with a tutorial introduction) in a separately submitted work.

Our software implementation of this algorithm is called ProtPal. We measured the accuracy of ProtPal relative to leading non-MCMC alignment/reconstruction protocols by simulating indels and substitutions on a known phylogeny, withholding the true history and attempting to reconstruct it from the sequences at the tips of the tree. The results show that all previous approaches to the reconstruction of ancestral sequences introduce significant biases, including systematic underestimation of insertions and overestimation of deletions. This contradicts previous claims that advances in the statistical foundations of alignment tools, supported by improvements in protein-structure benchmarks, necessarily improve the accuracy of evolutionary parameter estimates like the indel rate [6, 22, 23].

ProtPal introduces less bias than any other methods we tested, including PRANK, the state-of-the-art phylogenetic progressive aligner [6]. Based on our

tests, ProtPal appears to be the best choice for small to moderately-sized analyses, such as a reconstruction of the history of proteins at the inter-species level in human evolutionary history. Using ProtPal to estimate indel rates for  $\sim 7,500$  human protein-coding gene families, we find that per-gene indel rates are approximately gamma-distributed, with 95% of genes experiencing a mean rate of less than 0.1 indel events per synonymous substitution event. We find that lengths of inserted and deleted sequences are comparably distributed, having medians 5 and 7, respectively. The human lineage appears to have experienced unusually many insertions since the human-mouse split. By mapping genes to Gene Ontology (GO) terms, we find that the 200 fastest-indel genes are enriched for regulatory and metabolic functions. Possible applications and extensions of our algorithm include phylogenetic placement, homology detection, and reconstruction of structured RNA.

## 2 Results

### 2.1 Computational reconstruction of simulated histories

We undertook to determine the ability of leading bioinformatics programs, including ProtPal, to characterize mutation event histories. We simulated indel histories on a tree, then attempted to reconstruct the MAP history,  $\hat{H}$ , using only knowledge of the sequences  $S$  and the phylogeny  $T$  (but not the sequence alignment). The history  $\hat{H}$  is the aligned set of observed extant and predicted ancestral sequences, such that insertion, deletion, and substitution events can be pinpointed to specific tree branches (though not to specific time points on those branches).

We then characterized the reconstruction quality both directly, by comparison of  $\hat{H}$  to the true  $H$ , and indirectly, by using  $\hat{H}$  to estimate  $\theta$ , the evolutionary

parameters:

$$\hat{\theta}_{\hat{H}} = \operatorname{argmax}_{\theta'} P(\theta' | \hat{H}, S, T) = \operatorname{argmax}_{\theta'} P(\hat{H}, S | T, \theta') \quad (1)$$

where the latter step assumes a flat prior,  $P(\theta') = \text{const.}$  We then compared the history-conditioned parameter estimate  $\hat{\theta}_{\hat{H}}$  to the true  $\theta$ .

This statistic is not without its problems. For one thing, we use an initial guess of  $\theta$  to estimate  $\hat{H}$ . Furthermore, for an unbiased estimate, we should sum over all histories, rather than conditioning on the MAP reconstructed history. This summing over histories would, however, require multiple expensive calculations of  $P(S|T, \theta)$ , where conditioning on  $\hat{H}$  requires only one such calculation. Furthermore, parameter estimation conditioned on a MAP-reconstructed history is the *de facto* method employed by large-scale genomics studies focusing on indels [24–27].

### 2.1.1 Simulation model parameters

The model parameters are  $\theta = (\lambda^i, \lambda^d, \mathbf{p}^i, \mathbf{p}^d, \mathbf{R})$ : the insertion and deletion rates  $(\lambda^i, \lambda^d)$ , indel length distributions  $(\mathbf{p}^i, \mathbf{p}^d)$  and substitution rate matrix  $(\mathbf{R})$ . Here we focus on the rates  $(\lambda^i, \lambda^d)$ .

As described in Appendix A we generated data using an external simulation tool, indel-seq-gen, varying insertion  $(\lambda^i)$ , deletion  $(\lambda^d)$  and substitution rates  $(r)$  over a range representative of per-gene rates in *Amniota* evolution (Figure 4). We varied indel rates (with  $\lambda^i = \lambda^d$ ) between 0.005 and 0.08 expected indels per unit time, estimating that this range accounts for 95% of human gene families. We left the substitution model  $(\mathbf{R})$  and indel length distributions  $(\mathbf{p}^i, \mathbf{p}^d)$  fixed, employing indel-seq-gen’s empirically-estimated values.

We performed simulations on mammalian, amniote and fruitfly phylogenies, using the taxa in those clades for which genomic sequence is actually available.

We found generally consistent results, with common trends that were most pronounced on the largest of the three trees that we used (the twelve sequenced *Drosophila* species [28]). In discussing the trends, we will refer specifically to the results on this largest of the trees.

### 2.1.2 Indel rate estimates

**Overall most accurate** We first set out to determine which program, when used to analyze a set of unaligned sequences, returns the indel rate estimate closest to the true rate.

We report the ratio of inferred rate to true rate for insertions  $\frac{\hat{\lambda}_H^i}{\lambda^i}$  and deletions  $\frac{\hat{\lambda}_H^d}{\lambda^d}$  in Figure 1, with each  $\hat{\lambda}_H^* \in \{\hat{\lambda}_H^i, \hat{\lambda}_H^d\}$  defined as  $\hat{\theta}_H$  in Equation 7. No parameter estimate derived from a computationally reconstructed history approaches the level of accuracy achieved using the true history (labeled “True simulated history” in Figure 1).

The results do not always concord with previous benchmarks that have measured accuracy using 3D structural alignments: for example, the FSA program, one of the most accurate aligners on structural benchmarks [23], performs poorly here. This discordance may be due to the fundamental differences between evolutionary and structural homology, and the metrics used to assess each. For instance, consider a region with many nearby and overlapping insertions and deletions. The spatial and temporal location of these insertion and deletion events (in particular, the pinpointing of events to branches on the tree) defines what the “perfect” evolutionary reconstruction is. In contrast, even given perfect knowledge of the insertion/deletion history, a “perfect” structural alignment depends only on the proteins at the tips of the tree, and this alignment could differ from the true evolutionary reconstruction.

Fundamentally, the difference between FSA and ProtPal is the underlying metric that is being optimized by each program: FSA attempts to maximize



a metric (AMA=Alignment Metric Accuracy) which is essentially “structural” (in the sense that it predicts how many residues would be correctly aligned in a pairwise alignment of two leaf-node sequences, as might be used in structure prediction by target-template alignment), while ProtPal attempts to maximize a “phylogenetic” metric (the probability of a given evolutionary history). The metric we have used in our benchmark (which counts correct reconstruction of the number of indel events on branches of the tree) is also “phylogenetic”. By contrast, Appendix B Figure 7 shows the programs’ ranking using the AMA metric. FSA performs well, with accuracy exceeding that of ProtPal in the highest indel rate category. This suggests that the differences between our evolutionary benchmark and previous benchmarks are not due to the data, but rather the types of metrics that are used to measure alignment accuracy; similarly, the differences between the leading programs are primarily due to what types of benchmark they are explicitly trying to perform well at.

All programs other than ProtPal display insertion-*versus*-deletion biases that are, to a varying degree, asymmetric. Typically, the asymmetry is that insertions are underrepresented and deletions overrepresented. ProtPal’s bias, which is generally less than the other programs, is also the most symmetric: reconstructed insertions and deletions are roughly equally reliable, with both slightly underestimated.

Over the distribution of human gene rates used by this benchmark, our phylogenetic likelihood approach, ProtPal, provides the most accurate reconstructions of both insertion and deletion counts. PRANK, which also uses a tree (but no likelihood), avoids insertion-deletion biases to a certain extent, although insertion rates are slightly underestimated relative to deletion rates. Since ProtPal’s MAP history estimation appears similar to the optimization algorithm of PRANK, we suspect that ProtPal’s marginally better performance

is due primarily to its main difference in implementation: ProtPal tracks an *ensemble* of possible reconstructions during progressive tree traversal (Section 4), whereas PRANK uses a single “current best guess.”

**Effect of indel rate variation** To investigate the effect of indel rate variation on estimation accuracy, we separate each program’s error distributions by indel rate (Figure 2). We find that all programs’ accuracy is strongly affected by the indel rate used in simulation.

As the true indel rate increases, most programs’ estimates drift towards  $\frac{\hat{\lambda}_H^*}{\lambda^*} \rightarrow 0$ . This is consistent with the so-called “gap attraction” effect, where indels that are nearby in sequence can be misinterpreted as substitution events [29]. Depending on the phylogenetic orientation of the events, estimated rates can be elevated or lowered, with different biases for insertion and deletion rates (Figure 3).

Gap attraction and other biases operate simultaneously, and are sometimes opposed. MUSCLE over-estimates the deletion rate under most conditions, but (consistent with a trend where programs have lower  $\frac{\hat{\lambda}_H^*}{\lambda^*}$  at higher indel rates) gets the deletion rate roughly correct in the highest-indel-rate category of our benchmark. However, the alignments produced by MUSCLE at high indel rates are no more “accurate” by pairwise metrics (Appendix B Figure 7). We conjecture that multiple, contradictory types of gap attraction are at work, e.g. Figures 3B and 3C.

After ProtPal, the two most accurate reconstruction methods are PRANK and ProbCons (the latter combined with a parsimonious indel reconstruction). ProbCons produces more reliable insertion estimates than PRANK in a broad range of benchmark categories, is tied with PRANK for deletion estimates, and appears robust to indel rate variation. PRANK performs slightly better than ProbCons in the slowest indel rate category we considered. ProtPal produces the

most reliable estimates overall, outperforming ProbCons in all but the fastest indel rate category, and PRANK in all but the slowest.

**Sensitivity to substitution rate** As compared to variation of simulated indel rate, variation of simulated substitution rate appears to have little effect on the accuracy of indel reconstruction (Appendix B Figure 8). One notable exception is FSA, which appears to be affected by the substitution rate more than the other programs. For example, when the simulated indel and substitution rates are both low, FSA is comparable to the most accurate of the other programs (ProtPal); but when the substitution rate is increased, FSA’s error is greater than the least accurate program (CLUSTALW). Errors in estimating the substitution rate are comparable among the programs tested, and are similarly correlated with the simulation indel rate (Appendix B Figure 9).

## 2.2 Reconstructed indel histories of human genes

We present here a comprehensive set of reconstructions accounting for the evolutionary history of individual codons in human genes. We used genes in the **Orthologous and Paralogous Transcripts in Clades (OPTIC)** database’s *Amniota* set, comprised of the 5 mammals *H. sapiens*, *M. musculus*, *C. familiaris*, *M. domestica*, *O. anatinus* and *G. gallus* as an outgroup [30]. Considering only those families with one unique ortholog per species (approximately 7,500 families), we combined tree branch statistics across genes, using the species tree in Appendix C Figure 11. Our reconstructions are available at [http://biowiki.org/~oscar/optic\\_reconstruction.tar](http://biowiki.org/~oscar/optic_reconstruction.tar), and we provide here various graphical summaries of *Amniota* evolutionary history. Several negative results stand in contrast to earlier-reported trends.

**Indel rates** Insertion and deletion rates are approximately gamma-distributed (Figure 4). Roughly 95% of genes have indel rates  $< 0.1$  indels per synonymous substitution.

**Phylogenetic origins** In our simulations, ProtPal pinpoints residues’ “branch of origin” more reliably than other tools, with a 93% accuracy rate (Appendix B Figure 6). Many codons appeared to have been inserted following the human-mouse split (Appendix C Figure 10)

**Branch-specific indel rates** Using our reconstructions to estimate the rates of indel mutations along specific tree branches, we find evidence of an elevated insertion rate in the human (black) branch, as well as on the the *Amniota* - *Australophenids* (pink) branch (Appendix C Figure 10).

**Amino acid distributions** Distributions over amino acids differ significantly between inserted, deleted and non-indel sequences (Appendix C Figure 12). In general, small residues are over-represented in insertions, in agreement with previous studies [31].

**Indel lengths** We find, contrary to a previous study in *Nematode* [32], that length distributions in the Amniotes are nearly identical between insertions and deletions (Appendix C Figure 13). The previously-reported result may be attributable to the deletion-biased nature of the methods used, particularly CLUSTALW and MUSCLE [32].

**Indel position** The position of indels within genes is highly biased towards the ends of genes, presumably in large part reflecting annotation error (Appendix C Figure 14). The bias is strongest for deletions at the N-terminus of

the gene, but both insertions and deletions are enriched in both C- and N-termini.

**Evolutionary context of indel SNPs** We find no general correlation between the indel rate for a gene and the number of indel polymorphisms recorded for that gene in dbSNP [33] (Appendix C Figure 15).

**Gene ontology indel rates** No Gene Ontology (GO) categories stand out as having significantly lowered or heightened indel rates in any of the three ontologies, contrasting with the reported results of a 2007 study using a smaller number of genes [31]. An enrichment analysis conducted with GOstat [34] showed that the 200 fastest evolving genes in our data are significantly enriched for regulatory and metabolic functions.

### 3 Discussion

We developed and analyzed a simulation benchmark that compares programs based on their reconstructions of evolutionary history, using instantaneous mutation rates representative of Amniote evolution. We tested several different tree topologies; results were similar on all trees, but most pronounced on the tree with the longest branch lengths. We find that most programs distort indel rate measurements, despite claims to the contrary. Moreover, the systematic bias varies significantly when the rates of substitutions and indels are varied within a biologically reasonable range. Many of the programs we rated have been ranked in the past, but using benchmarks that use protein structural alignments as a gold standard, rather than evolutionary simulations. Furthermore, these previous benchmarks have not directly assessed the reconstruction of evolutionary history (or summary statistics such as the indel rate), but have used

other alignment accuracy metrics such as the *Sum of Pairs Score*. Alignment programs that perform weakly on our benchmark have apparently performed well on these previous benchmarks. We hypothesize that these benchmarks, compared to ours, are less directly predictive of a program’s accuracy at historical reconstruction, although they may better reflect the program’s suitability to assist in tasks relating more closely to folded structure, like prediction of a protein’s 3D structure from a homologous template. We have introduced a new notation that describes a general, hidden Markov model-structured likelihood function for indel histories on a tree, as well as the structure of the corresponding inference algorithm. We have implemented the new method in a freely-available program, ProtPal, that allows, for the first time, phylogenetic reconstruction with accuracy over a broad range of indel rates. ProtPal is written in C++ as a part of the DART package: [www.biowiki.org/ProtPal](http://www.biowiki.org/ProtPal). The evolutionary reconstructions ProtPal produces are, according to our simulated tests, the most accurate of any available tool, for a range of parameters typical of human genes.

We applied ProtPal to the reconstruction of human gene indel history, using families of human gene orthologs from the OPTIC database. We find some patterns that agree with previous studies, such as the non-uniform distributions over amino acids seen in [31]. Other results stand in contrast - a previous study found significantly different length distributions for insertions and deletions [32], whereas in our data they appear very similar. Another prediction of our reconstruction is an elevated rate of insertions on the human branch since the human-mouse split. This contrasts with a previous analysis [35], though the data therein was whole genomes, rather than individual protein-coding genes. In contrast to [31], we find no obvious predictive power of the Gene Ontology (GO) for indel rates; that is, the indel rate does not appear strongly correlated with the presence or absence of any particular GO term-gene association. How-

ever, enrichment analysis for GO terms using Gostat [34] showed that the 200 fastest-evolving genes are significantly enriched for regulatory and metabolic function. This apparent discrepancy might be explained by a group of regulatory and metabolic genes which have very high indel rates, but whose small number prevent them from skewing the average within their GO categories.

Many applications which use a fixed-alignment phylogenetic likelihood could potentially benefit from ProtPal’s reconstruction profiles. For example, phylogenetic placement algorithms estimate taxonomic distributions by evaluating the relative likelihoods of placing sequence reads on tree branches [36]. By using sequence profiles exported from ProtPal, these reads could be placed with greater attention to indels and a more realistic accounting for alignment uncertainty. Homology detection could be done in a similar way, thereby making use of the phylogenetic relationship of the sequences within the reference family. It has been observed that the detection of positive selection is highly sensitive to the alignment used [7]. ProtPal could be modified to detect selection using entire profiles rather than single alignments, potentially eliminating the bias brought on by an inaccurate alignment.

In summary, multiple alignments are frequently constructed for use in downstream evolutionary analyses. However, except for our method and slow-performing MCMC methods, there are no software tools for reconstructing molecular evolutionary history that explicitly maximize a phylogenetic likelihood for indels. Our results strongly indicate that algorithms such as ProtPal (which use such a phylogenetic model) produce significantly more reliable estimates of evolutionary parameters, which we believe to be highly indicative of evolutionary accuracy. These results falsify previous assertions that existing, non-phylogenetic tools are well-suited to this purpose. Furthermore, we have demonstrated that it is possible to achieve such accuracy without sacrificing asymptotic guarantees

on time/memory complexity, or resorting to expensive MCMC methods. ProtPal can reconstruct phylogenetic histories of entire databases on commodity hardware, enabling the large-scale study of evolutionary history in a consistent phylogenetic framework.

## 4 Methods

The details concerning generation and analysis of simulated data are contained in Appendix A. A mathematically complete description of the alignment algorithm has been submitted as a separate work, and an early version has been made available online here: <http://arxiv.org/abs/1103.4347>.

### 4.1 Felsenstein’s algorithm for indel models

Our algorithm may be viewed as a generalization of Felsenstein’s pruning recursion [19], a widely-used algorithm in bioinformatics and molecular evolution. A few common applications of this algorithm include estimation of substitution rates [37]; reconstruction of phylogenetic trees [38]; identification of conserved (slow-evolving) or recently-adapted (fast-evolving) elements in proteins and DNA [39]; detection of different substitution matrix “signatures” (e.g. purifying vs diversifying selection at synonymous codon positions [40], hydrophobic vs hydrophilic amino acid signatures [41], CpG methylation in genomes [42], or basepair covariation in RNA structures [43]); annotation of structures in genomes [44,45]; and placement of metagenomic reads on phylogenetic trees [36].

Felsenstein’s algorithm computes  $P(S|T, \theta)$  for a substitution model by tabulating intermediate probability functions of the form  $G_n(x) = P(S_n|x_n = x, \theta)$ , where  $x_n$  represents the individual residue state of ancestral node  $n$ , and  $S_n$  represents all the sequence data that is causally descended from node  $n$  in the tree (i.e. the observed residues at the set of leaf nodes whose most recent common



ancestor is node  $n$ ).

The pruning recursion visits all nodes in postorder. Each  $G_n$  function is computed in terms of the functions  $G_l$  and  $G_r$  of its immediate left and right children (assuming a binary tree):

$$\begin{aligned} G_n(x) &= P(S_n | x_n = x, \theta) \\ &= \begin{cases} \left( \sum_{x_l} M_{x, x_l}^{(l)} G_l(x_l) \right) \left( \sum_{x_r} M_{x, x_r}^{(r)} G_r(x_r) \right) & \text{if } n \text{ is not a leaf} \\ \delta(x = S_n) & \text{if } n \text{ is a leaf} \end{cases} \end{aligned}$$

where  $M_{ab}^{(n)} = P(x_n = b | x_m = a)$  is the probability that node  $n$  has state  $b$ , given that its parent node  $m$  has state  $a$ ; and  $\delta(x = S_n)$  is a Kronecker delta function terminating the recursion at the leaf nodes of the tree. These  $G_n$  functions are often referred to as “messages” in the machine-learning literature [46].

Our new algorithm is algebraically equivalent to Felsenstein’s algorithm, if the concept of a “substitution matrix” over a particular alphabet is extended to the countably-infinite set of all sequences over that alphabet. Our chosen class of “infinite substitution matrix” is one that has a finite representation: namely, the *finite-state transducer*, a probabilistic automaton that transforms an input sequence to an output sequence, and a familiar tool of statistical linguistics [47].

By generalizing the idea of matrix multiplication ( $AB$ ) to two transducers ( $A$  and  $B$ ), and introducing a notation for feeding the same input sequence to two transducers in parallel ( $A \circ B$ ), we are able to write Felsenstein’s algorithm in a new form (see Section 4.3):

$$G_n = \begin{cases} (M^{(l)} G_l) \circ (M^{(r)} G_r) & \text{if } n \text{ is not a leaf} \\ \nabla(S_n) & \text{if } n \text{ is a leaf} \end{cases}$$

where  $\nabla(S_n)$  is the transducer equivalent of the Kronecker delta  $\delta(x = S_n)$ .

The function  $G_n$  is now encapsulated by a transducer “profile” of node  $n$ .

This representation has complexity  $\mathcal{O}(L^N)$  for  $N$  sequences of length  $L$ , which we reduce to  $\mathcal{O}(LN)$  by stochastic approximation of the  $G_n$ . This approximation relies on the *alignment envelope* [48], a data structure introduced by prior work on efficient alignment methods. The alignment envelope is a subset of all the possible histories in which most of the probability mass is concentrated. A related data structure is the *partial order graph* [21]. Both these data structures can be viewed as ensembles of possible histories, in contrast to a single “best-guess” reconstruction of the history. Figure 5 shows a state graph, with paths through it corresponding to histories relating the two sequences GL and GIV. The paths highlighted in blue form a partial order graph, corresponding to a subset of these histories generated by a stochastic traceback. At each progressive traversal step, we sample a high-probability subset of alignments of two sibling profiles in order to maintain a bound on the state space size. Note that if we sample only the most likely path at every internal node, we essentially recover the progressive algorithm of PRANK, and if we sample and store all solutions, we recover the machine  $G_n$  with state space of size  $\mathcal{O}(L^N)$ .

## 4.2 Transducer definitions and lemmas

The definitions and lemmas are presented in a condensed form here, and expanded upon in [49].

A transducer is a tuple  $(\Omega, \Psi, \Phi, \phi_S, \phi_E, \tau, \mathcal{W})$  where  $\Omega$  is an input alphabet,  $\Psi$  is an output alphabet,  $\Phi$  is a set of states,  $\phi_S \in \Phi$  is the start state,  $\phi_E \in \Phi$  is the end state,  $\tau \subseteq \Phi \times (\Omega \cup \{\epsilon\}) \times (\Psi \cup \{\epsilon\}) \times \Phi$  is the transition relation, and  $\mathcal{W} : \tau \rightarrow [0, \infty)$  is the transition weight function.

Suppose that  $T = (\Omega, \Psi, \Phi, \phi_S, \phi_E, \tau, \mathcal{W})$  and  $U = (\Omega', \Psi', \Phi', \phi'_S, \phi'_E, \tau', \mathcal{W}')$  are transducers.

Let  $\mathcal{W}(\pi)$  be the product of all transition weights along a state path  $\pi$  and let  $\mathcal{W}(x : [T] : y)$  be the sum of such weights for all paths whose input labels, concatenated, yield the string  $x \in \Omega^*$  and whose output labels yield  $y \in \Psi^*$ .

*Equivalence:* If  $T$  and  $U$  have the same input and output alphabets ( $\Omega = \Omega'$  and  $\Psi = \Psi'$ ) and the same sequence weights  $\mathcal{W}(x : [T] : y) = \mathcal{W}'(x : [U] : y) \forall x, y$ , then we say the transducers are *equivalent*,  $T \equiv U$ . Less formally, we will write  $T \cong U$  if  $\mathcal{W}(x : [T] : y) \simeq \mathcal{W}'(x : [U] : y)$ .

*Moore transducers:* The *Moore normal form* for transducers, named for Moore machines [50], associates input/output with three distinct types of state: *Match*, *Insert* and *Delete*. Paths through Moore transducers can be associated with (gapped) pairwise alignments of input and output sequences. For any transducer  $T$ , there exists an equivalent Moore-normal form transducer  $U$  with  $|\Phi'| = \mathcal{O}(|\tau|)$  and  $|\tau'| = \mathcal{O}(|\tau|)$ .

*Composition:* If  $T$ 's output alphabet is the same as  $U$ 's input alphabet ( $\Psi = \Omega'$ ), there exists a transducer,  $TU = (\Omega, \Psi', \Phi'' \dots \mathcal{W}'')$ , that unifies the output of  $T$  with the input of  $U$ , such that  $\forall x \in \Omega^*, z \in (\Psi')^*$ :

$$\mathcal{W}''(x : [TU] : z) = \sum_{y \in \Psi^*} \mathcal{W}(x : [T] : y) \mathcal{W}'(y : [U] : z) \quad (2)$$

If  $T$  and  $U$  are in Moore form, then  $|\Phi''| \leq |\Phi| \times |\Phi'|$  and  $|\tau''| \leq |\tau| \times |\tau'|$ .

*Intersection:* If  $T$  and  $U$  have the same input alphabets ( $\Omega = \Omega'$ ), there exists a transducer,  $T \circ U = (\Omega, \Psi'', \Phi'' \dots \mathcal{W}'')$ , that unifies the input of  $T$  with the input of  $U$ . The output alphabet is  $\Psi'' = (\Psi \cup \{\epsilon\}) \times (\Psi' \cup \{\epsilon\})$ , i.e. a  $T$ -output symbol (or a gap) aligned with a  $U$ -output symbol (or a gap).

Let  $\text{alignments}(t, u) \subset (\Psi'')^*$  denote the set of all gapped pairwise alignments of sequences  $t \in \Psi^*$  and  $u \in (\Psi')^*$ . Transducer  $T \circ U$  has the property

that  $\forall x \in \Omega^*, t \in \Psi^*, u \in (\Psi')^*$ :

$$\sum_{v \in \text{alignments}(t, u)} \mathcal{W}''(x : [T \circ U] : v) = \mathcal{W}(x : [T] : t) \mathcal{W}'(x : [U] : u) \quad (3)$$

If  $T$  and  $U$  are in Moore form, then  $|\Phi''| \leq |\Phi| \times |\Phi'|$  and  $|\tau''| \leq |\tau| \times |\tau'|$ . Paths through  $T \circ U$  are associated with three-way alignments of the input sequence to the two output sequences.

*Identity:* Let  $\mathcal{I}$  be a transducer that copies input to output unmodified, so  $\mathcal{I}T \equiv T\mathcal{I} \equiv T$ .

*Exact match:* For any sequence  $S \in \Omega^*$ , there exists a Moore-form transducer  $\nabla(S) = (\Omega, \emptyset, \Phi, \tau \dots)$  with  $|\Phi| = \mathcal{O}(\text{length}(S))$  and  $|\tau| = \mathcal{O}(\text{length}(S))$ , that rejects all input except  $S$ , such that  $\mathcal{W}(x : [\nabla(S)] : \epsilon) = 1$  if  $x = S$ , and 0 if  $x \neq S$ . Note that  $\nabla(S)$  outputs nothing (the empty string).

*Chapman-Kolmogorov transducers:* A transducer  $T$  is *probabilistic* if  $\mathcal{W}(x : [T] : y)$  represents a probability  $P(y|x, T)$ : that is, for any given input string,  $x$ , it defines a probability measure on output strings,  $y$ .

Suppose  $T(t)$  is a function returning a probabilistic transducer of the form  $(\Omega, \Omega, \Phi, \phi_S, \phi_E, \tau, \mathcal{W}(t))$ , i.e. a transducer whose transition weight  $\mathcal{W}$  depends on an additional *time parameter*,  $t$ , and which satisfies the transducer equivalence  $T(t)T(t') \equiv T(t + t') \forall t, t'$ .

Then  $T(t)$  gives the finite-time transition probabilities of a homogeneous continuous-time Markov process on the strings  $\Omega^*$ , as the above transducer equivalence is a form of the Chapman-Kolmogorov equation.

If the state space of  $T$  is finite, then this equation describes a renormalization of the composed state space  $\Phi \times \Phi$  back down to the original state space  $\Phi$ . So far, only one nontrivial time-dependent transducer is known that solves this equation exactly using a finite number of states: the TKF91 model [51].

### 4.3 The phylogenetic likelihood

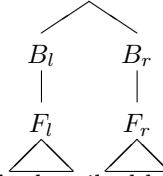
We rewrite the evidence,  $P(S|T, \theta)$  for sequences  $S$ , tree  $T$ , and parameters  $\theta$ , in the form  $P(\{S_n : n \in \mathcal{L}\} | R, \{B_n\})$  where  $\{S_n : n \in \mathcal{L}\}$  denotes the set of sequences observed at leaf nodes,  $\{B_n\}$  denotes the stochastic evolutionary processes occurring on the branches, and  $R$  denotes the probabilistic model for the sequence at the root node of the tree.

The root and branch transducers  $(R, \{B_n\})$  represent an alternative view of the tree and parameters  $(T, \theta)$ . The root transducer  $R$  outputs from the equilibrium or other initial distribution of the process. If  $(p, c) \in T$  is a parent-child pair, then  $B_c = B(T_{pc})$  is a time-dependent transducer parameterized by the branch length. In practise, the branch transducers need not satisfy the Chapman-Kolmogorov equation for the following constructs to be of use; for example, the  $\{B_n\}$  might be approximations to true Chapman-Kolmogorov transducers [52].

Let  $R = (\emptyset, \Omega, \dots)$  be a transducer outputting sequences sampled from the prior at the phylogenetic root.

Let  $n$  be a tree node. If  $n$  is a leaf, define  $F_n = \mathcal{I}$ . Otherwise, let  $(l, r)$  denote the left and right child nodes, and define  $F_n = (B_l F_l) \circ (B_r F_r)$  where  $B_n = (\Omega, \Omega, \dots)$  is a transducer modeling the evolution on the branch leading to  $n$ .

Diagrammatically we can write  $F_n$  as



The phylogenetic likelihood is then fully described by  $F = R F_{\text{root}}$ .

Like  $R$ , transducer  $F$  models a probability distribution over output sequences, but accepts only the empty string as an input sequence. This empty

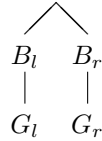
input sequence is just a technical formality (transducers must have inputs); if we ignore it, we can think of  $F$  and  $R$  as hidden Markov models (HMMs), rather than transducers.  $R$  is an HMM that generates a single sequence,  $F$  a multi-sequence HMM that generates the whole set of leaf sequences.

Inference with HMMs often uses a dynamic programming matrix (e.g. the Forward matrix) to track the ways that a given evidential sequence can be produced by a given grammar.

For our purposes it is useful to introduce the evidence in a different way, by transforming the model to incorporate the evidence directly. We augment the state space so that the model is no longer capable of generating any sequences *except* the observed  $\{S_n\}$ , by composing  $F_{\text{root}}$ 's forked outputs with exact-match transducers that will only accept the observed sequences at the leaves of the tree. This yields a model,  $G$ , whose state space is of size  $\mathcal{O}(L^N)$  and, in fact, is directly analogous to the Forward matrix.

If  $n$  is a leaf node, then let  $G_n = \nabla(S_n)$  where  $S_n$  is the sequence at  $n$ . Otherwise,  $G_n = (B_l G_l) \circ (B_r G_r)$ .

Diagrammatically we can write  $G_n$  as



Let  $G = R G_{\text{root}}$ . The evidence is  $P(\{S_n\} | R, \{B_n\}) = \mathcal{W}(\epsilon : [G] : \epsilon)$ .

The net output of  $G$  is always the empty string. The sequences  $\{S_n\}$  are recognized as inputs by the  $\nabla(S_n)$  transducers at the tips of the tree, but are not passed on as outputs themselves.

Likewise, the input of  $G$  is the empty string, because  $R$  accepts only the empty string on its input.

We can think of  $G$  as a Markov model, rather than an HMM. It has no input or output; rather, the sequences are encoded into its structure.

Transducer  $G$  has  $\mathcal{O}(L^N)$  states, which is impractically many, so ProtPal uses a progressive hierarchy  $H_n$  of approximations to the corresponding  $G_n$ , with state spaces that are bounded in size.

If  $n$  is a leaf node, let  $H_n = \nabla(S_n) = G_n$ . Otherwise, let  $H_n = (B_l E_l) \circ (B_r E_r)$  where  $\Phi_{E_n} \subseteq \Phi_{H_n}$  is a subset defined by sampling complete paths through the Markov model  $M_n = RH_n$  and adding the  $H_n$ -states used by those paths to  $\Phi_{E_n}$ , until the pre-specified bound on  $|\Phi_{E_n}|$  is reached. Then  $G \cong M_{\text{root}}$ .

The likelihood of a given history may be calculated by summing over paths through  $G$  consistent with that history. In the simplest cases (e.g. minimal Moore-form branch transducers), each indel history corresponds to exactly one path, so the MAP indel history corresponds to the maximum-weight state path through  $G$ .

#### 4.4 Alignment envelopes

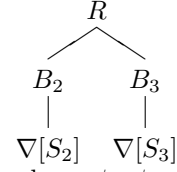
Let  $\nabla(S)$  be defined such that it has only one nonzero-weighted path

$$X_0 \rightarrow W_0 \xrightarrow{S_1} M_1 \rightarrow W_1 \xrightarrow{S_2} M_2 \rightarrow \dots \rightarrow W_{L-1} \xrightarrow{S_L} M_L \rightarrow W_L \rightarrow X_L$$

so a  $\nabla(S)$ -state is either the start state ( $X_0$ ), the end state ( $X_L$ ), a wait state ( $W_i$ ) or a match state ( $M_i$ ). All these states have the form  $\phi_i$  where  $i$  represents the number of symbols of  $S$  that have to be read in order to reach that state, i.e. a “co-ordinate” into  $S$ . All  $\nabla(S)$ -states are labeled with such co-ordinates, as are the states of any transducer that is a composition involving  $\nabla(S)$ , such as  $G_n$  or  $H_n$ .

For example, in a simple case involving a root node (1) with two children (2,3) whose sequences are constrained to be  $S_2, S_3$ , the evidence transducer is

$$G = RG_{\text{root}} = R(G_2 \circ G_3) = R((B_2 \nabla(S_2)) \circ (B_3 \nabla(S_3))) =$$



All states of  $G$  have the form  $g = (r, b_2, \phi_2 i_2, b_3, \phi_3 i_3)$  where  $\phi_2, \phi_3 \in \{X, W, M\}$ , so  $\phi_2 i_2 \in \{X_{i_2}, W_{i_2}, M_{i_2}\}$  and similarly for  $\phi_3 i_3$ . Thus, each state in  $G$  is associated with a co-ordinate pair  $(i_2, i_3)$  into  $(S_2, S_3)$ , as well as a state-type pair  $(\phi_2, \phi_3)$ .

Let  $n$  be a node in the tree, let  $\mathcal{L}_n$  be the set of indices of leaf nodes descended from  $n$ , and let  $G_n$  be the phylogenetic transducer for the subtree rooted at  $n$ , defined in Section 4.3. Let  $\Phi_n$  be the state space of  $G_n$ .

If  $m \in \mathcal{L}_n$  is a leaf node descended from  $n$ , then  $G_n$  includes, as a component, the transducer  $\nabla(S_m)$ . Any  $G_n$ -state,  $g \in \Phi_n$ , is a tuple, one element of which is a  $\nabla(S_m)$ -state,  $\phi_i$ , where  $i$  is a co-ordinate (into sequence  $S_m$ ) and  $\phi$  is a state-type. Define  $i_m(g)$  to be the co-ordinate and  $\phi_m(g)$  to be the corresponding state-type.

Let  $A_n : \Phi_n \rightarrow 2^{\mathcal{L}_n}$  be the function returning the set of *absorbing leaf indices* for a state, such that the existence of a finite-weight transition  $g' \rightarrow g$  implies that  $i_m(g) = i_m(g') + 1$  for all  $m \in A_n(g)$ .

Let  $(l, r)$  be two sibling nodes. The *alignment envelope* is the set of sibling state-pairs from  $G_l$  and  $G_r$  that can be aligned. The function  $E : \Phi_l \times \Phi_r \rightarrow \{0, 1\}$  indicates membership of the envelope. For example, this basic envelope allows only sibling co-ordinates separated by a distance  $s$  or less

$$E_{\text{basic}}(f, g) = \max_{m \in A_l(f), n \in A_r(g)} |i_m(f) - i_n(g)| \leq s \quad (4)$$

An alignment envelope can be based on a *guide alignment*. For leaf nodes  $x, y$  and  $1 \leq i \leq \text{length}(S_x)$ , let  $\mathcal{G}(x, i, y)$  be the number of residues of sequence  $S_y$  in the section of the guide alignment from the first column, up to and including



the column containing residue  $i$  of sequence  $S_x$ .

This envelope excludes a pair of sibling states if they include a homology between residues which is more than  $s$  from the homology of those characters contained in the guide alignment:

$$E_{\text{guide}}(f, g) = \max_{m \in A_l(f), n \in A_r(g)} \max( |\mathcal{G}(m, i_m(f), n) - i_n(g)|, |\mathcal{G}(n, i_n(g), m) - i_m(f)| ) \leq s \quad (5)$$

Let  $K(x, i, y, j)$  be the number of match columns (those columns of the guide alignment in which both  $S_x$  and  $S_y$  have a non-gap character) between the column containing residue  $i$  of sequence  $S_x$  and the column containing residue  $j$  of sequence  $S_y$ . This envelope excludes a pair of sibling states if they include a homology between residues which is more than  $s$  matches from the homology of those characters contained in the guide alignment:

$$E_{\text{guide}}(f, g) = \max_{m \in A_l(f), n \in A_r(g)} \max( |\mathcal{G}(m, i_m(f), n) - K(m, i_m(f), n, i_n(g))|, |\mathcal{G}(n, i_n(g), m) - K(n, i_n(g), m, i_m(f))| ) \leq s$$

## 4.5 OPTIC data analysis

**Data** Amniote gene families were downloaded from <http://genserv.anat.ox.ac.uk/downloads/clades/>. We restricted our analysis to the  $\sim 7,500$  families having simple 1:1 orthologies. The same species tree topology (downloaded from <http://genserv.anat.ox.ac.uk/clades/amniota/displayPhylogeny>) was used for all reconstructions, though branch lengths were estimated separately for each family as part of OPTIC. When computing branch-specific indel rates, the branch lengths of the species tree were used.

**Reconstruction and rate estimation** Gene families were aligned and reconstructed using ProtPal with a 3-rate class Markov chain over amino acids, insertion and deletion rates set to 0.01, and 250 traceback samples. Averaged and per-branch indel rates were computed with ProtPal using the `-pi` and `-pb` options. The indel rates were then normalized by the synonymous substitution rate for each corresponding nucleotide alignment (taken directly from OPTIC), computed with PAML [53]. Residues' origins were determined by finding the tree node closest to the root containing a non-gap reconstructed character.

**External data** Genes were mapped to Gene Ontology terms via the mapping downloaded from [http://www.ebi.ac.uk/GOA/human\\_release.html](http://www.ebi.ac.uk/GOA/human_release.html) during 10/2010. Indel SNPs per gene were taken from a table downloaded from Supplemental Table 5 of [54].

## 5 Figures

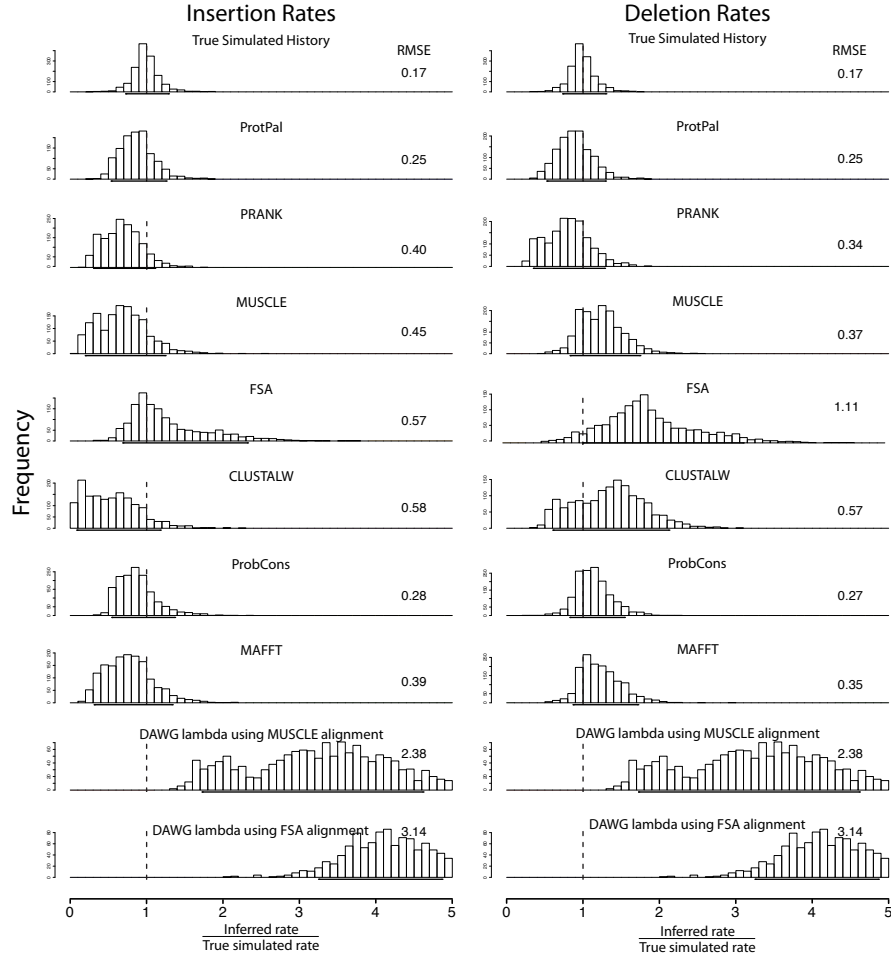


Figure 1: ProtPal’s estimates of insertion and deletion rates are the most accurate of any program tested, as measured by the RMSE of  $\frac{\hat{\lambda}_H}{\lambda^*}$  values aggregated over all substitution/indel rate categories. Quantiles containing 90% of the data are shown as a bolded portion of the  $x$ -axis, and RMSE is shown to the right of each distribution, the latter computed as described in Appendix A Equation 6. No aligner approaches the accuracy of the rates estimated with the true alignment, though ProtPal, PRANK, and ProbCons are the top three, with ProtPal as the most accurate over all. Many aligners, particularly MUSCLE, CLUSTALW, and MAFFT, significantly underestimate insertion rates and overestimate deletion rates. ProtPal and PRANK perform their own ancestral reconstruction and other alignment programs were augmented with a most-recent-common-ancestor (MRCA) parsimony as described in [55].

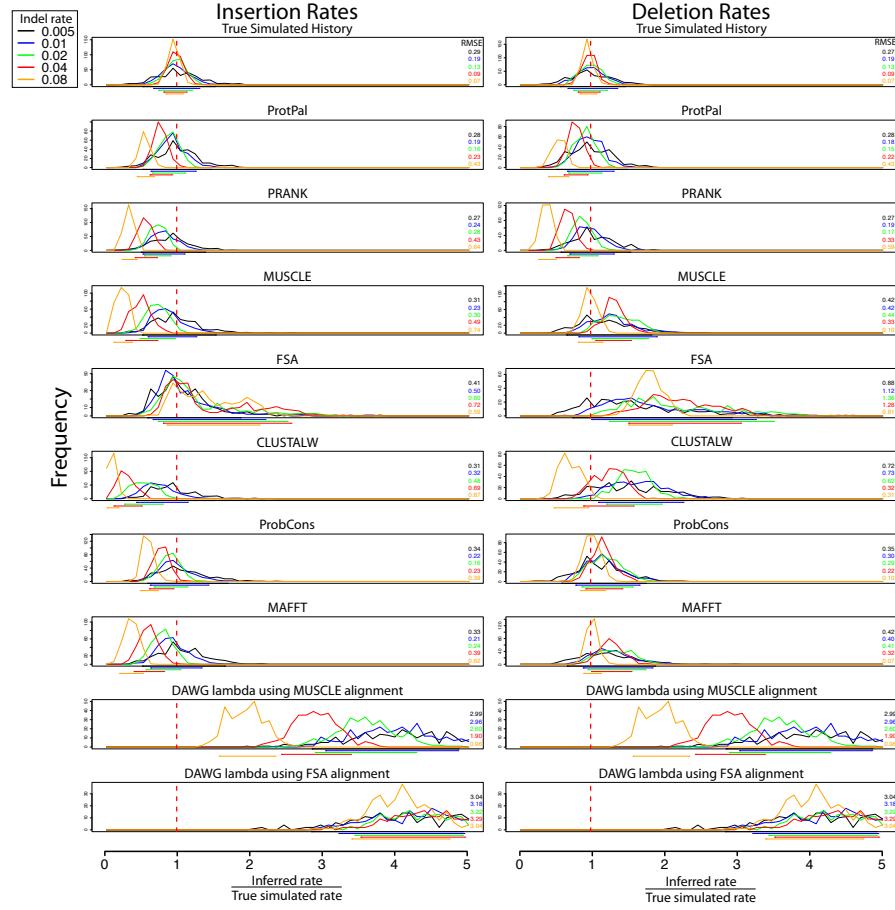


Figure 2: Rate estimation accuracy is highly dependent on the simulated indel rate. For instance, PRANK is more accurate at lower indel rates, ProbCons is more accurate at higher rates. ProtPal is more accurate than PRANK in all but one rate (0.005) and equal or more accurate than ProbCons in all but one rate (0.08). The drift towards  $\frac{\text{inferred}}{\text{true}} = 0$  exhibited by most programs indicates that most programs infer proportionally fewer indels as rates are increased, likely due to various forms of gap attraction. Color-coded 90% quantiles and RMSEs are shown underneath and to the right of each group of distributions, respectively. RMSE is computed as described in Appendix A Equation 6

## 6 References

### References

1. X. Qu, R. Swanson, R. Day, J. Tsai, *Curr Protein Pept Sci.* **10**, 270 (2009).
2. A. M. Moses, D. Y. Chiang, D. A. Pollard, V. N. Iyer, M. B. Eisen, *Genome Biology* **5** (2004).
3. K. S. Pollard, *et al.*, *Nature* **443**, 167 (2006).
4. J. D. Thompson, F. Plewniak, O. Poch, *Nucleic Acids Research* **27**, 2682 (1999).
5. K. M. Wong, M. A. Suchard, J. P. Huelsenbeck, *Science* **319**, 473 (2008).
6. A. Löytynoja, N. Goldman, *Science* **320**, 1632 (2008).
7. P. Markova-Raina, D. Petrov, *Genome Research* **21**, 863 (2011).
8. S. Nelesen, K. Liu, D. Zhao, C. R. Linder, T. Warnow, *Pacific Symposium on Biocomputing* **2008**, 25 (2008).
9. K. Liu, S. Nelesen, S. Raghavan, C. R. Linder, T. Warnow, *IEEE/ACM Trans Comput Biol Bioinform* **6**, 7 (2009).
10. E. project consortium, *Genome Research* **17**, 760 (2007).
11. R. K. Bradley, *et al.*, *PLoS ONE* **4**, e6478 (2009).
12. C. Strobe, K. Abel, S. Scott, E. Moriyama, *Mol Biol Evol* **26**, 2581 (2009).
13. I. Holmes, W. J. Bruno, *Bioinformatics* **17**, 803 (2001).
14. M. A. Suchard, B. D. Redelings, *Bioinformatics* **22**, 2047 (2006).

15. A. Löytynoja, N. Goldman, *Proceedings of the National Academy of Sciences of the USA* **102**, 10557 (2005).
16. R. C. Edgar, *BMC Bioinformatics* **5**, 113 (2004).
17. M. Larkin, *et al.*, *Bioinformatics* **23**, 2947 (2007).
18. K. Katoh, K. Kuma, H. Toh, T. Miyata, *Nucleic Acids Research* **33**, 511 (2005).
19. J. Felsenstein, *Journal of Molecular Evolution* **17**, 368 (1981).
20. D. G. Higgins, A. J. Bleasby, R. Fuchs, *Computer Applications in the Biosciences* **8**, 189 (1992).
21. C. Lee, C. Grasso, M. Sharlow, *Bioinformatics* **18**, 452 (2002).
22. R. A. Cartwright, *Bioinformatics* **21 Suppl 3**, iii31 (2005).
23. R. K. Bradley, *et al.*, *PLoS Computational Biology* **5**, e1000392 (2009).
24. O. Kamneva, A. Liberles, N. Ward, *Genome Biology and Evolution* **2**, 870 (2010).
25. Z. Zhang, J. Huang, Z. Wang, L. Wang, G. Peiji, *Molecular Biology and Evolution* **28**, 291 (2011).
26. L. Zhu, Q. Wang, P. Tang, H. Araki, D. Tian, *Molecular Biology and Evolution* **26**, 2353 (2009).
27. L. Gomez-Valero, *et al.*, *Molecular Ecology* **17**, 4382 (2008).
28. A. G. Clark, *et al.*, *Nature* **450**, 203 (2007).
29. G. Lunter, *Bioinformatics* **23**, 289 (2007).
30. A. Heger, C. Ponting, *NAR* **36**, 267 (2008).

31. N. de la Chaux, P. Messeer, P. Arndt, *BMC Evolutionary Biology* **7** (2007).
32. Z. Wang, *et al.*, *BMC Evol Biol.* **9** (2009).
33. S. Saccone, *et al.*, *Nucleic Acids Res* (2011).
34. T. Beissbarth, T. P. Speed, *Bioinformatics* **20**, 1464 (2004).
35. *Nature* (2002).
36. F. A. Matsen, R. B. Kodner, E. V. Armbrust, *BMC Bioinformatics* **11**, 538 (2010).
37. Z. Yang, *Journal of Molecular Evolution* **39**, 105 (1994).
38. B. Rannala, Z. Yang, *Journal of Molecular Evolution* **43**, 304 (1996).
39. A. Siepel, D. Haussler, *Journal of Computational Biology* **11**, 413 (2004).
40. Z. Yang, R. Nielsen, N. Goldman, A.-M. Pedersen, *Genetics* **155**, 432 (2000).
41. J. L. Thorne, N. Goldman, D. T. Jones, *Molecular Biology and Evolution* **13**, 666 (1996).
42. A. Siepel, D. Haussler, *Molecular Biology and Evolution* **21**, 468 (2004).
43. B. Knudsen, J. Hein, *Bioinformatics* **15**, 446 (1999).
44. A. Siepel, D. Haussler, *Proceedings of the eighth annual international conference on research in computational molecular biology, San Diego, March 27-31 2004*, P. Bourne, D. Gusfield, eds. (ACM, 2004), pp. 177–186.
45. J. S. Pedersen, *et al.*, *PLoS Computational Biology* **2**, e33 (2006).

46. F. R. Kschischang, B. J. Frey, H.-A. Loeliger, *IEEE Transactions on Information Theory* **47**, 498 (1998).
47. M. Mohri, F. Pereira, M. Riley, *Computer Speech and Language* **16**, 69 (2002).
48. B. Paten, *et al.*, *Genome Research* **18**, 1829 (2008).
49. O. Westesson, G. Lunter, B. Paten, I. Holmes, *arXiv* (2011). ArXiv:1103.4347v1.
50. E. F. Moore, *Gedanken-experiments on Sequential Machines* (Princeton University Press, Princeton, N.J., 1956), vol. 34 of *Annals of Mathematical Studies*, chap. 5, pp. 129–153.
51. J. L. Thorne, H. Kishino, J. Felsenstein, *Journal of Molecular Evolution* **33**, 114 (1991).
52. I. Miklós, G. Lunter, I. Holmes, *Molecular Biology and Evolution* **21**, 529 (2004).
53. Z. Yang, *Molecular Biology and Evolution* **24**, 1586 (2007).
54. R. Mills, *et al.*, *Genome Research* **16** (2006).
55. S. Sinha, E. Siggia, *MBE* **22** (2005).
56. C. B. Do, M. Brudno, S. Batzoglou, PROBCONS: Probabilistic consistency-based multiple alignment of amino acid sequences (2004). Submitted.
57. I. Holmes, *A DART tutorial*, Berkeley Drosophila Genome Project, LSA Room 539, UC Berkeley (2000). A tutorial for probabilistic methods and hidden Markov models, presented with the aid of the author’s software



package implementing many common HMM algorithms. Available from <http://www.fruitfly.org/~ihh/>.

## A Simulation parameters and setup

**Data generation** Our simulation study is comprised of alignments simulated using 5 different indel rates (0.005, 0.01, 0.02, 0.04, and 0.08 indels per unit time), each with 3 different substitution rates (0.5, 1, and 2 expected substitutions per unit time) and 100 replicates. Time is defined such that a sequence evolving for time  $t$  with substitution rate  $r$  is expected to accumulate  $rt$  substitutions per site. We employed an independent third-party simulation program, *indel-seq-gen*, specifically designed to generate realistic protein evolutionary histories [12]. *indel-seq-gen* is capable of modeling an empirically-fitted indel length distribution, rate variation among sites, and a neighbor-aware distribution over inserted sequences allowing for small local duplications. Since the indel and substitution model used by *indel-seq-gen* are separate from (and richer than) those used by ProtPal, ProtPal has no unfair advantage in this test.

*indel-seq-gen* v2.0.6 was run with the following command:

```
cat guidetree.tree| indel-seq-gen -m JTT -u xia --num_gamma_cats 3 -a
0.372 --branch_scale r/b --outfile simulated_alignment.fa --quiet --outfile_format
f -s 10000 --write_anc
```

The above command uses the “JTT” substitution model, the “xia” indel fill model (based on neighbor effects, estimated from E coli k-12 proteins [12]), and 3 gamma-distributed rate categories with shape 0.372. Branch lengths are scaled by the substitution rate for simulation rate  $r$ , normalized by the inverse of *indel-seq-gen*’s underlying substitution rate ( $b = 1.2$ ) so as to adhere to the above definition of evolutionary “time”. Similarly, indel rates, which are set in the guide tree file *guidetree.tree*, are scaled by  $\frac{b}{r}$  so that  $t\lambda^*$  insertions/deletions

are expected over time  $t$  for rate  $\lambda^*$ .

The root mean squared error (RMSE) for each error distribution was computed as follows:

$$RMSE = \sqrt{\sum_{replicates} \left( \frac{\hat{\lambda}_{\hat{H}}^*}{\lambda^*} - 1 \right)^2} \quad (6)$$

The true tree was made available to all programs which can utilize a tree (ProtPal, PRANK, MUSCLE), representing the use case in which the true tree is known (e.g. via the species tree) but the true alignment is unknown. We ran simulations on three different phylogenies: a tree of twelve sequenced *Drosophila* genomes [28] and trees from the mammalian and amniotic clades of the OPTIC database. We here report results for the *Drosophila* tree, which we empirically observe to show trends consistent with, but more pronounced than, those of the mammalian and amniotic trees. The clearer trends may be due to the *Drosophila* tree being larger than the other trees (12 taxa), or having a diverse range of branch lengths (0.001 - 0.59 expected substitutions/site, at the genome-wide average rate). The simulation data, reconstructions, and analysis scripts are available from [http://biowiki.org/~oscar/simulation\\_reconstruction.tar](http://biowiki.org/~oscar/simulation_reconstruction.tar).

**Alignment** We investigated several multiple alignment tools [15–18, 23, 56] in combination with alignment-conditioned reconstruction methods. Programs were run with their default settings, with the exception of PRANK and MUSCLE. To specify ancestral inference, the guide tree, and “insertions opening forever”, PRANK used the extra options “`-writeanc -t <treefile> +F`”. PRANK’s `-F` option allows insertions to match characters at alignments closer to the root. This can be a useful heuristic safeguard when an incorrect tree may produce errors in subtree alignments that cannot be corrected at internal nodes closer to the root. Since the true guide tree is provided to PRANK, it is safe to treat insertions in a strict phylogenetic manner via the `+F` option. For computational

efficiency, ProtPal was provided with a CLUSTALW guide alignment. Any alignment of the sequences can be used as a guide, and we chose CLUSTALW for its general poor performance, so that ProtPal would gain no unfair advantage by the information contained in the guide alignment. MUSCLE was provided the guide tree with the additional option “-usetree <treefile>”.

### **Muscle v3.6**

```
MUSCLE -in unaligned.fa -out aligned.fa -usetree guidetree.tree
```

### **PRANK v.080820**

```
PRANK -d=unaligned.fa -noxml -realbranches -writeanc -o=output_directory  
-t=guidetree.tree +F
```

### **Clustal v2.03**

```
clustalw -INFILE=unaligned.fa -OUTFILE=aligned.fa
```

### **ProbCons v1.12**

```
probcons unaligned.fa > aligned.fa
```

### **FSA v1.08**

```
fsa unaligned.fa > aligned.fa
```

### **MAFFT v6.818b**

```
mafft unaligned > aligned.fa
```

```
muscle 3.6 -in <infile> -out <outfile> -usetree <guide tree>
```

```
prank v.
```

```
cluswal 2.03 $(clustalw) -INFILE=$< -OUTFILE=$@.clustalw
```

```
probcons 1.12 probcons <infile>
```

```
fsa 1.08 fsa <infile>
```

```
mafft v6.818b mafft <infile>
```

**Imputing indel histories** The ancestral reconstruction programs ProtPal and PRANK were used to directly impute indel histories. The remaining tools were augmented to reconstruction tools by post-processing their MSAs using the maximum parsimony algorithm described in [55], with the ambiguous cases described therein (e.g. where a column of characters could be equally parsimoniously explained by a deletion on one child branch or an insertion on the other) resolved by a uniformly random choice from the possible solutions. Indel rates were estimated by counting indel events in MAP reconstructed histories:

$$\hat{\theta}_{\hat{H}} = \operatorname{argmax}_{\theta'} P(\theta' | \hat{H}, S, T) = \operatorname{argmax}_{\theta'} P(\hat{H}, S | T, \theta') \quad (7)$$

where the latter step assumes a flat prior,  $P(\theta') = \text{const.}$

This statistic is not without its problems. For one thing, we use an initial guess of  $\theta$  to estimate  $\hat{H}$ . Furthermore, for an unbiased estimate, we should sum over all histories, rather than conditioning on the MAP reconstructed history. This summing over histories would, however, require multiple expensive calculations of  $P(S|T, \theta)$ , where conditioning on  $\hat{H}$  requires only one such calculation. We further justify our benchmark of parameter estimates conditioned on a MAP-reconstructed history by noting that this the *de facto* method employed by large-scale genomics studies focusing on indels [24–27].

As well as imputing indel rates from reconstructed histories, we also tried using the `lambda.pl` program in the DAWG package [22], which estimates indel rates from MSAs directly (without attempting reconstruction).

**Estimating substitution rates** Substitution rates were estimated for each inferred alignment using XRate’s built-in EM algorithm and the following simple rate matrix. Given an equilibrium distribution over amino acid characters, with  $\pi_i$  defining the proportion of character  $i$ , the rate of character  $i$  mutating to  $j$

is set to  $r\pi_j$  where  $r$  is the only free rate parameter. XRate’s estimate of  $r$  is taken to be the average substitution rate of the MSA.

By using indel-seq-gen’s branch-scale option and changing the indel rate parameters accordingly, we are able to modulate the substitution and indel rates independently in the data generation step. This true substitution rate and the rate inferred by XRate are then directly comparable.

## **B Supplemental Figures: simulated data analysis**

## **C Supplemental Figures: OPTIC analysis**

In addition to estimating indel rates for all genes in the OPTIC set, we performed various other analyses which were left out of the main text for reasons of space limitations. We provide figures those displaying results here.

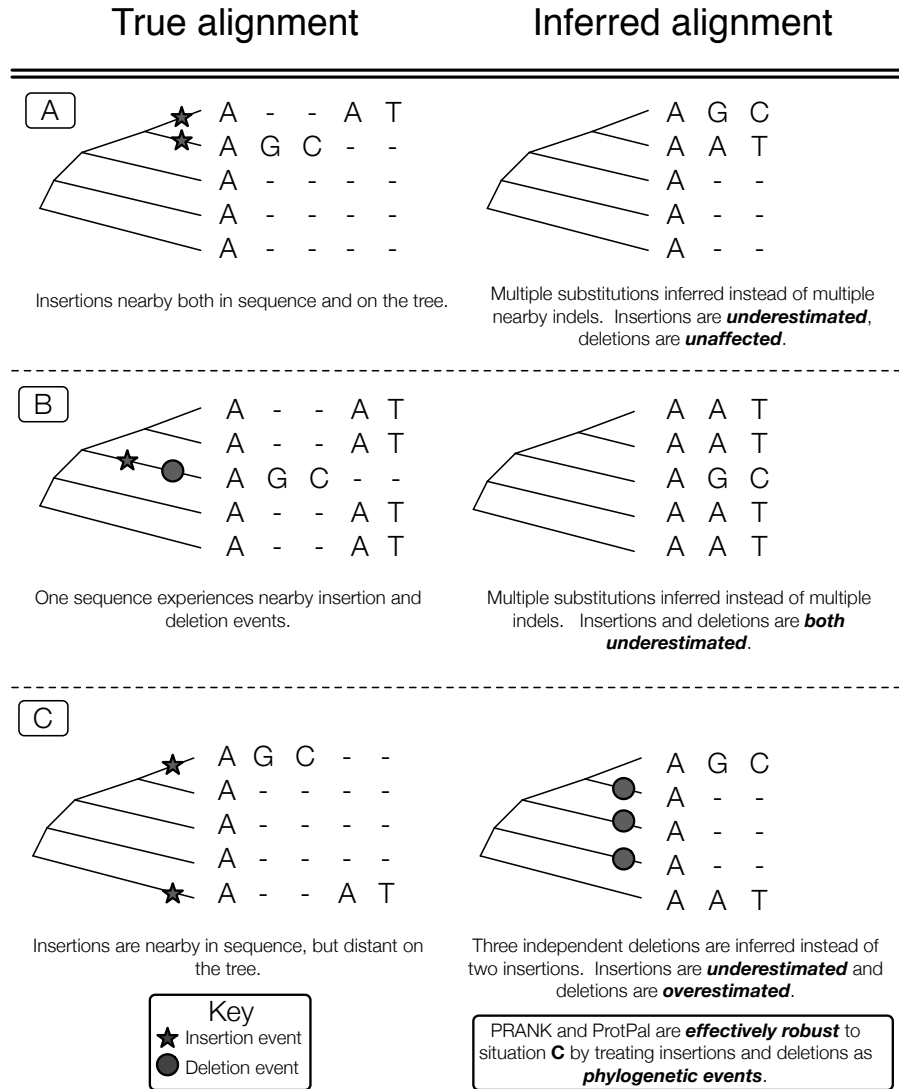


Figure 3: *Gap attraction*, the canceling of nearby complementary indels, can affect insertion and deletion rates in various ways depending on the phylogenetic relationship of the sequences involved. All programs are, to some extent, sensitive to situations **A** and **B** whereas phylogenetic aligners can avoid situation **C**. An insertion at a leaf requires gaps at all other leaves - an understandably costly alignment move when gaps are added without regard to the phylogeny, resulting in **multiple penalization** for each insertion. Such a penalization would cause most non-phylogenetic aligners to prefer the “Inferred alignment” in case **C** where there are fewer total gaps. Aligners treating indels as phylogenetic events would penalize each of the implied multiple deletions and only penalize each insertion once, thus preferring the “True alignment” in case **C**.

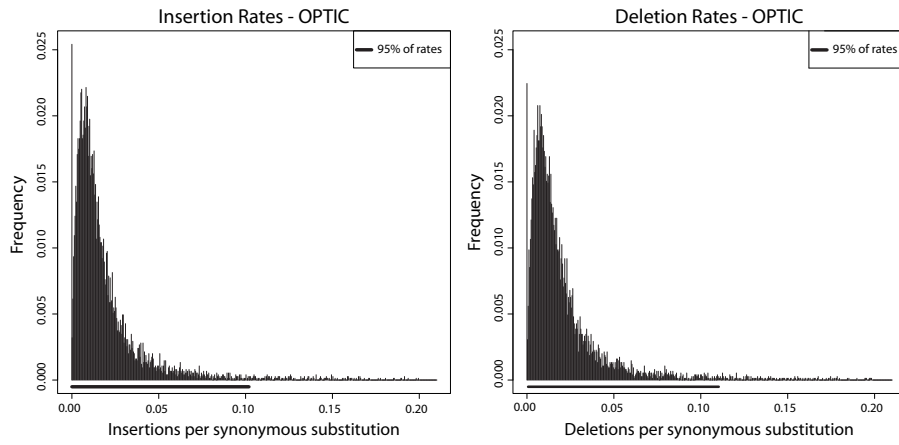


Figure 4: Insertion and deletion rates in *Amniota* show similar distributions, with 95% of genes having rates less than approximately 0.1 indels per synonymous substitution. Insertion and deletion rates were estimated using reconstructions done with ProtPal from a set of approximately 7,500 protein-coding genes from the OPTIC amniote database [30]. Indel rates were normalized by the synonymous substitution rate of each gene as computed with PAML [53] so that the plotted rate represents the number of expected indels per synonymous substitution. Since these rates are conditioned on the MAP reconstructed history, there are many alignments whose inferred indel rates are zero (197, 174, and 54 for insertions, deletions, and both, respectively).



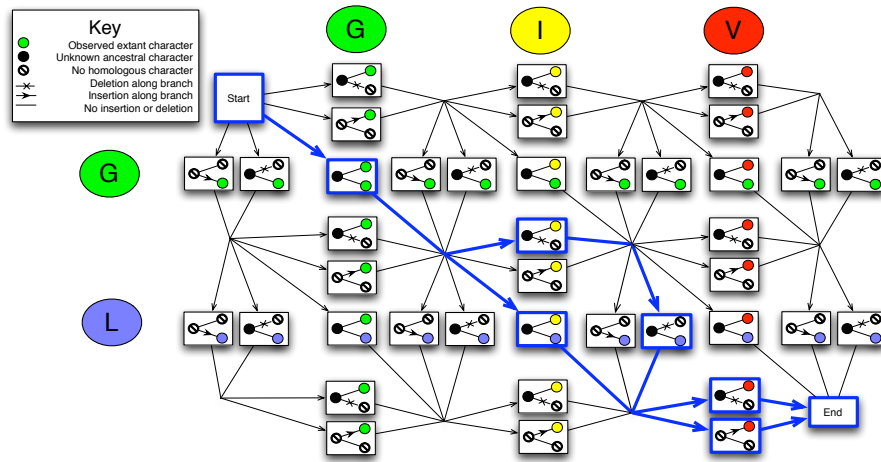


Figure 5: Each path through this state graph represents a possible evolutionary history relating sequences GL and GIV. By using stochastic traceback algorithms (sampling paths proportional to their posterior probability, blue highlighted states and transitions), it is possible to select a high-probability subset of the full state graph. By constructing such a subset at each internal node, it is possible to maintain a bound on the state space size during progressive tree traversal while still retaining an ensemble of possible histories.

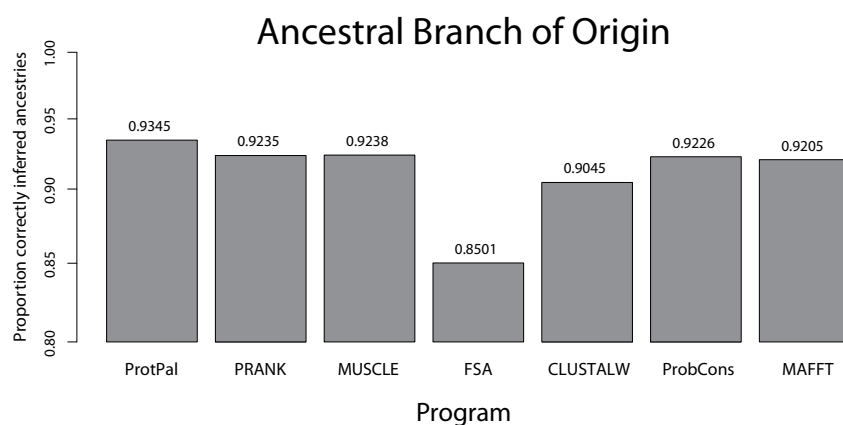


Figure 6: ProtPal correctly reconstructs the age of more extant residues than any other program tested. The  $y$ -axis shows the proportion of extant residues whose point of origin on the phylogenetic tree was correctly pinpointed by the reconstruction. The branch of origin was found by taking the tree node closest to the root containing a non-gap reconstructed character. All programs except FSA are in the 92%-94% range, owing to the fact that many columns (especially at low indel rates) are devoid of indels, making inference of origin trivial (as these columns' origin is pre-root).

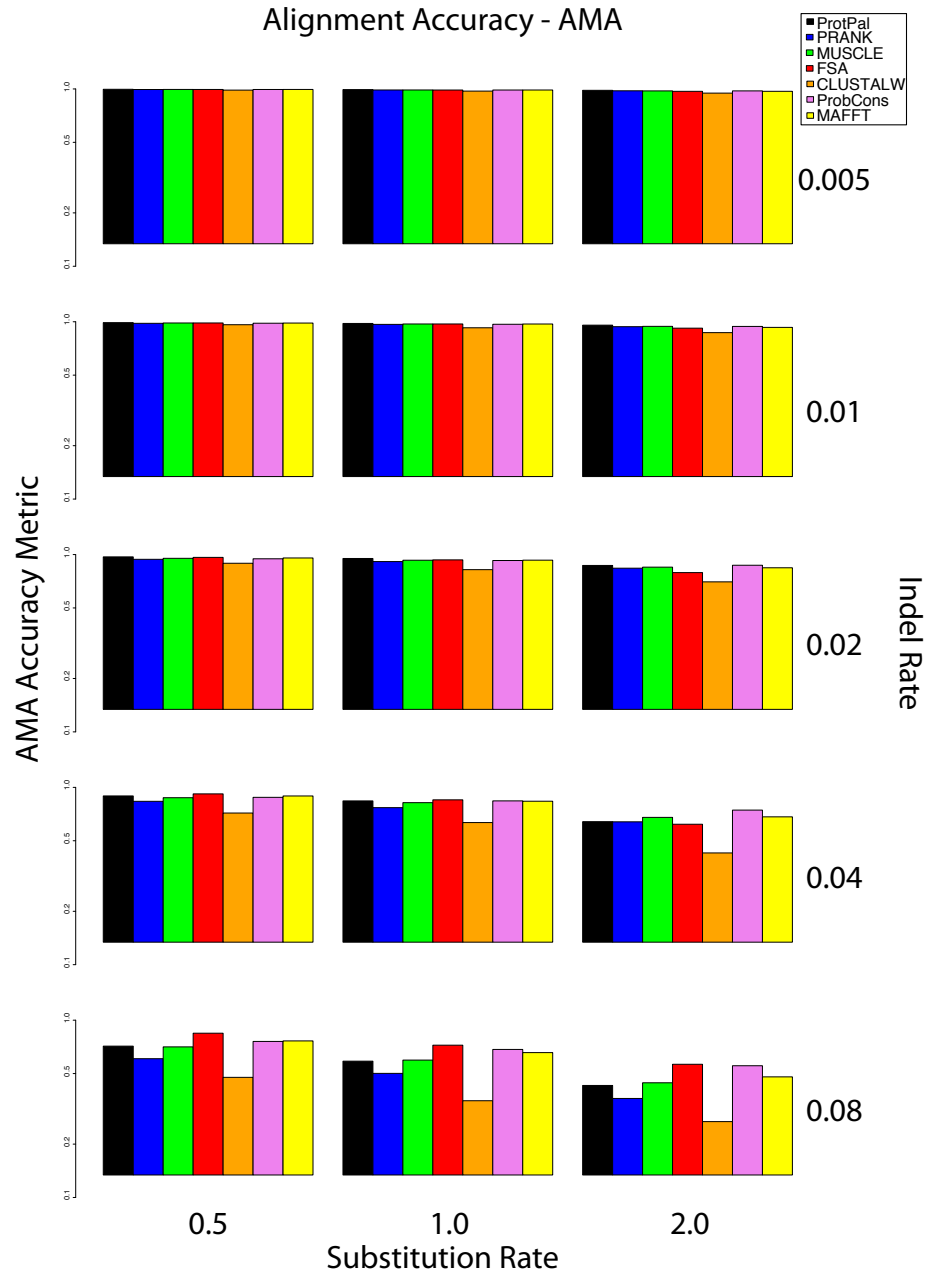


Figure 7: Cross-comparison of AMA scores and rate estimation accuracy reveals that using a single metric to assess alignment accuracy can be unreliable. AMA scores were computed for each programs alignment of only leaf sequences using **cmpalign** from the DART package [57]. AMA scores are comparable across programs until higher indel rates, where FSA performs best—contrasting with Figures 1 and 2 (main text). MUSCLE’s accurate deletion rate measurements at high rates and the low corresponding AMA scores suggest a “cancellation of biases”.

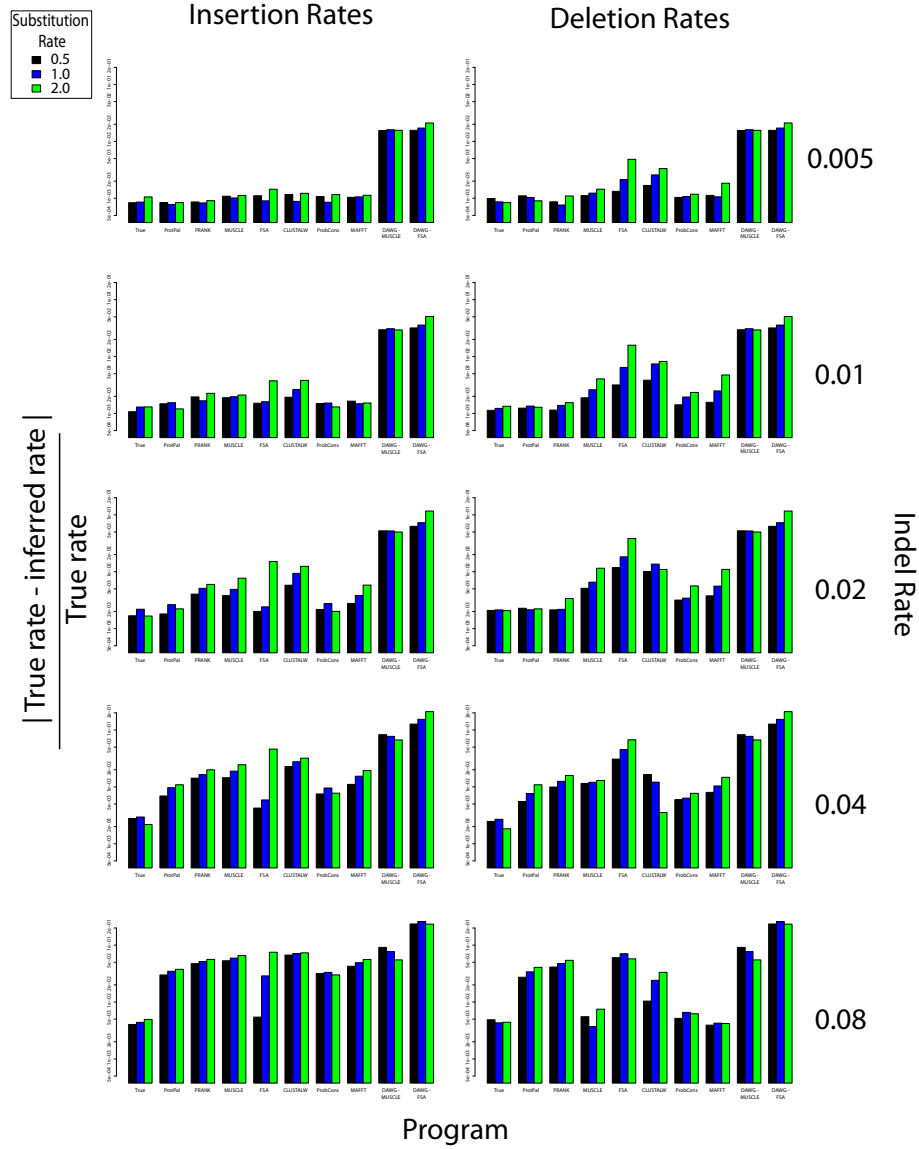


Figure 8: Most programs are relatively robust to variations in the simulated substitution rate, as evidenced by the benchmark data grouped according to substitution rate. Accuracy of rate estimation is plotted as  $|true - inferred|$  on the  $y$ -axis, with bars grouped by program for each indel rate and 3-tuple of substitution rates. Higher substitution rates often lead to higher error, presumably because they obscure homologies, making it more difficult to distinguish substitutions from indels. FSA appears more sensitive to increased substitutions than other programs - at indel rate 0.02, FSA's insertion rates are as accurate as ProtPal's at 0.5 and 1.0 substitutions per site, whereas at the highest substitution rate (2.0), its error exceeds that of CLUSTALW.

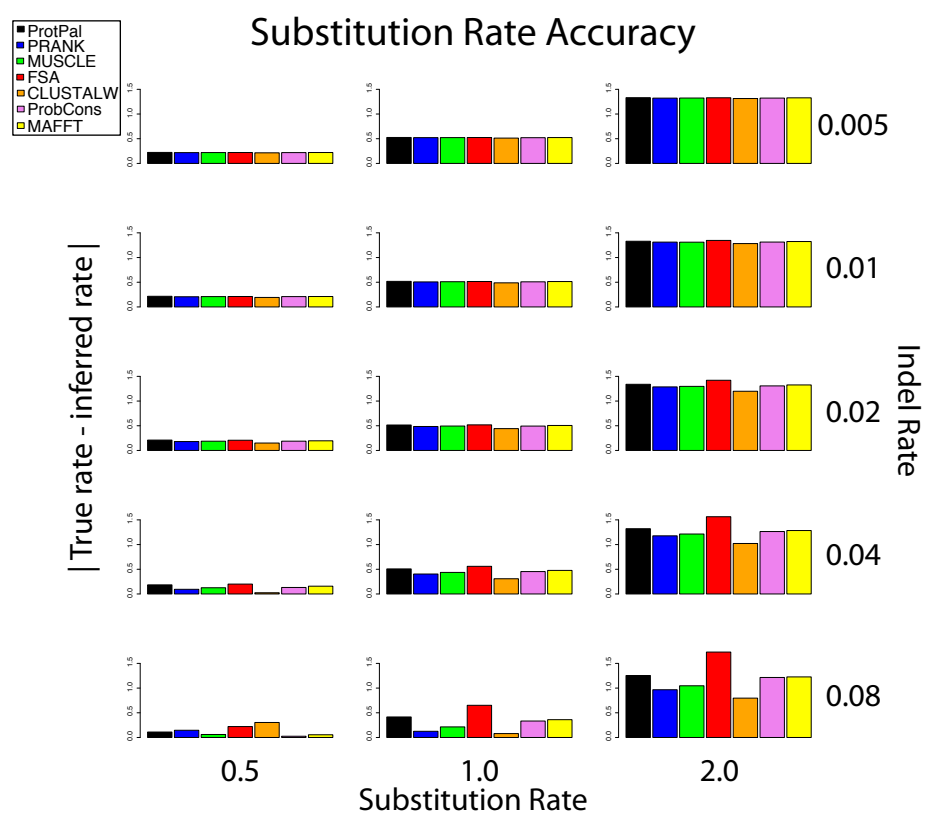


Figure 9: Substitution rates estimated from multiple alignments display comparable accuracy across methods.

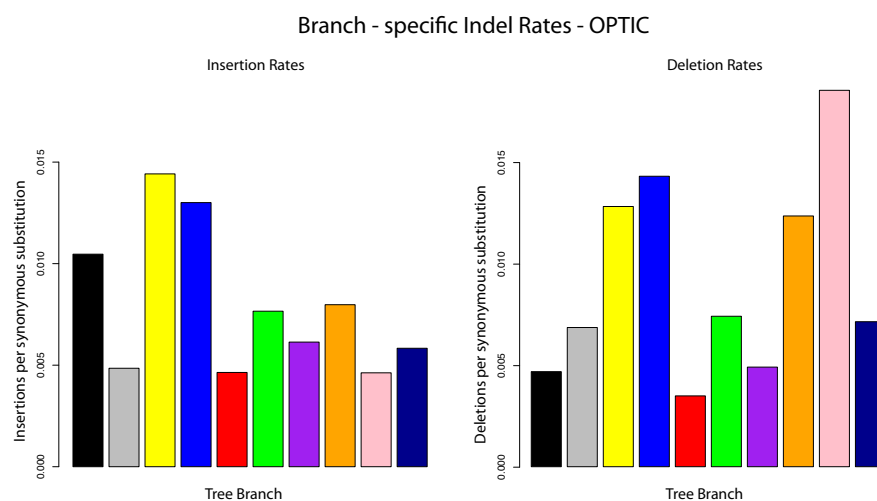


Figure 10: Reconstruction allows for estimation of branch-specific indel rates, revealing possibly interesting signals of evolution. Indel rates were averaged over all alignments, using the species tree shown in Figure 11. The human branch (*Euarhontoglires* - *H.sapiens*) appears to have experienced unusually many insertions. The *Amniota* - *Australophenids* (pink) branch has a higher deletion than insertion rate, though it is difficult to distinguish an insertion on this branch from a deletion on the *Amniota* - *G.gallus* (navy) branch. All other branches are comparable between insertions and deletions. Each bar is colored according to branches in Figure 11.

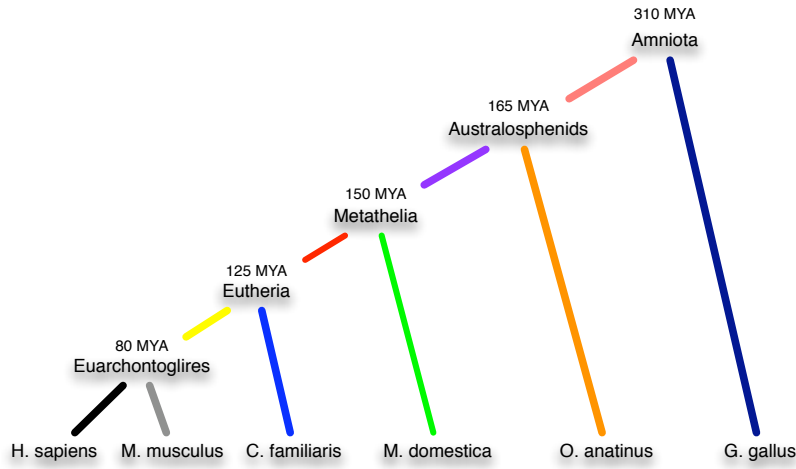


Figure 11: The phylogenetic tree used for analysis of OPTIC data, colored to inform the branch-specific Figure 10.

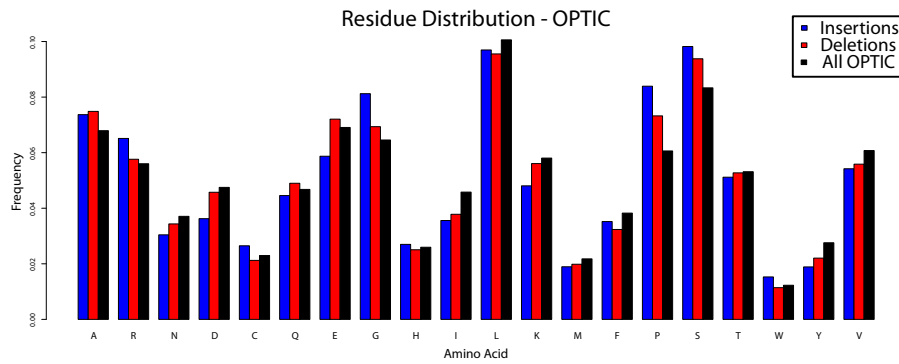


Figure 12: Distributions over amino acids are highly non-uniform, and differ between insertions, deletions, and the overall distribution seen in OPTIC. Inserted, deleted, and all sequences were separately pooled across all OPTIC genes reconstructed and amino acid distributions were computed for each.

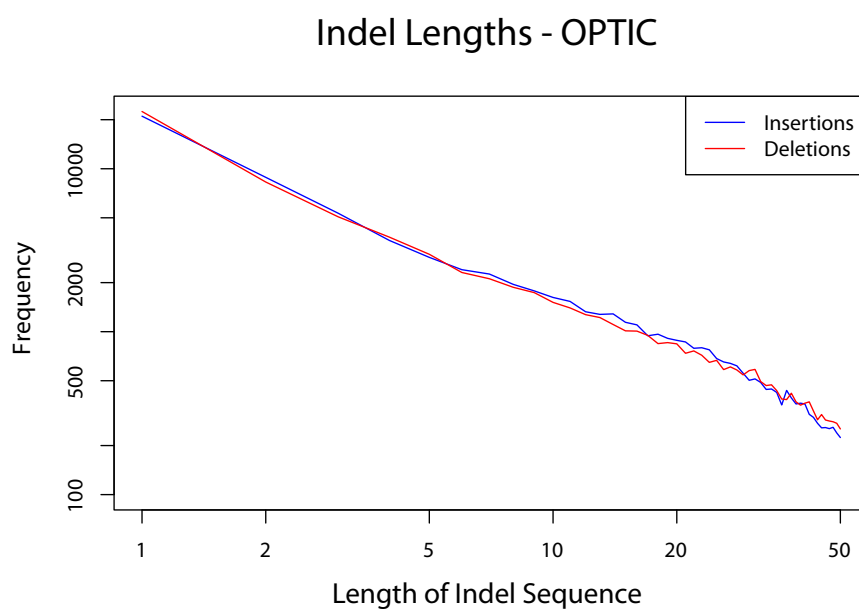


Figure 13: Lengths of inserted and deleted sequences are similarly distributed, in contrast to the conclusions of previous studies, such as [32], which found that deletions were longer relative to insertions in *C. elegans* sequence data. While this may represent a genuine difference in the evolution of human and worm genomes, it is likely that the use of deletion-biased aligners (MUSCLE and CLUSTALW) affected their conclusions.



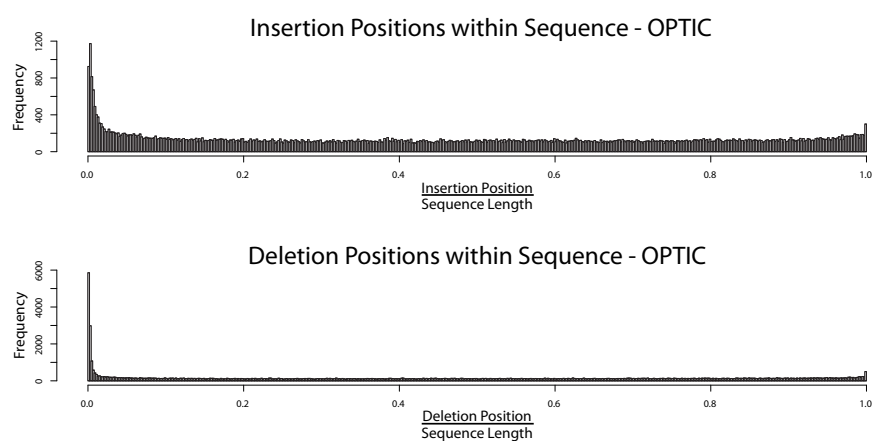


Figure 14: Indels are highly non-uniform in their distribution across genes: we see a 6-fold enrichment for insertions within the N-terminal 1% of the protein sequence, and a 1.4-fold increase within the C-terminal 1%. There is an 14-fold enrichment in deletions within the N-terminal 1% of the protein sequence, and a 1.8-fold increase within the C-terminal 1%. Indel locations are normalized by gene length to enable combining data across all OPTIC genes analyzed. This may be a mix of genuine biology (e.g. indels occur more often near the ends of genes) and artifacts (annotation errors are more likely to occur at the ends of genes).

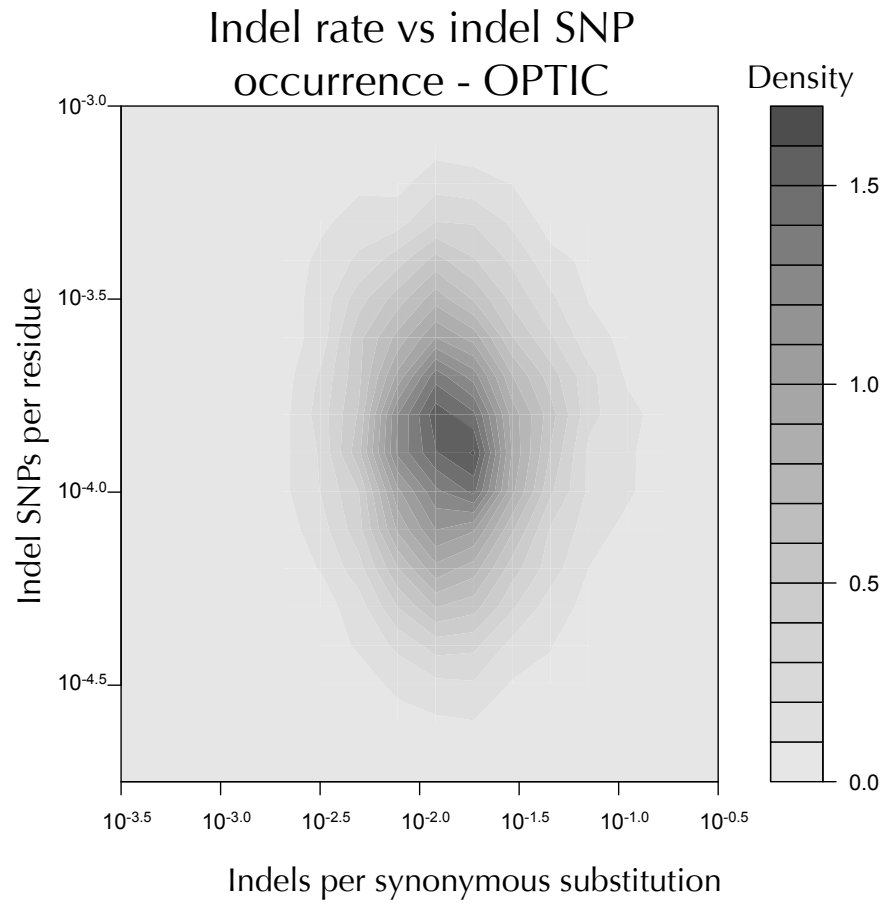


Figure 15: Visualizing the number of indel SNPs per residue (using only human sequence) against the evolutionary indel rate (computed across the *Amniote* clade) shows no correlation.